Azzolini Riccardo 2020-04-16

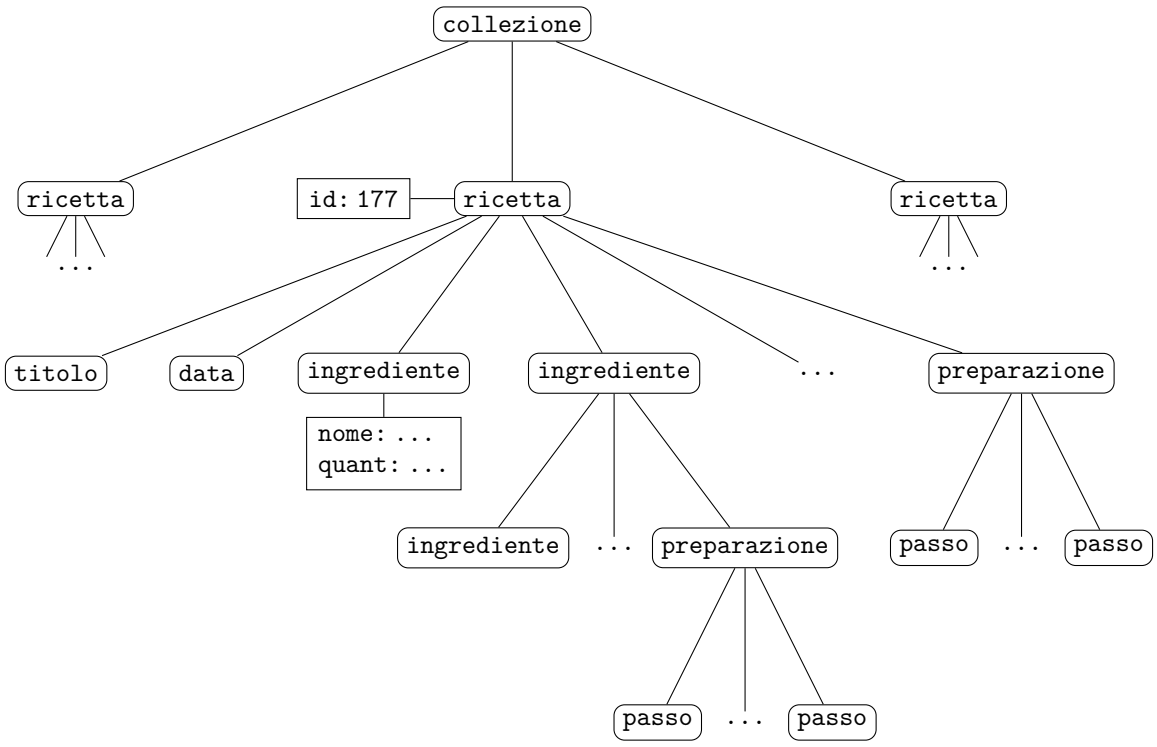
XPath

1 Percorsi di locazione

Per formare un percorso di locazione, si concatenano una sequenza di passi di locazione, separati dal carattere `/`. Ad esempio, il seguente percorso è costituito da tre passi:

```
child::ricetta[attribute::id='117']
/ child::ingrediente
/ attribute::quant
```

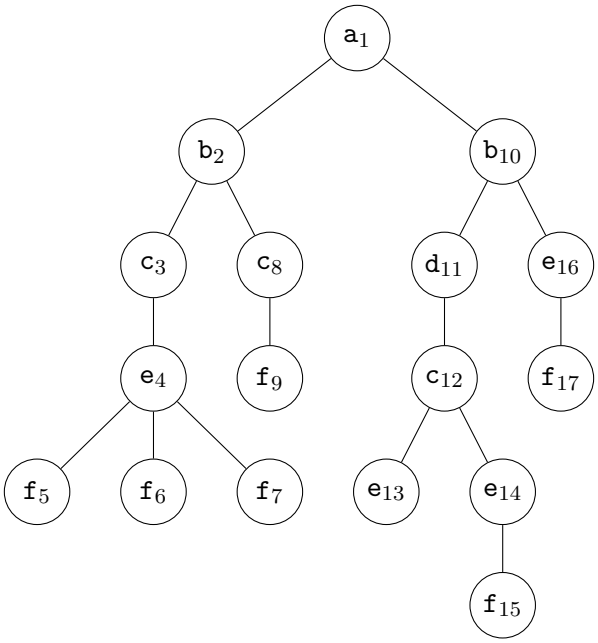
Supponendo di considerare il solito albero XML delle ricette,



con la radice (`collezione`) come nodo contesto, la valutazione di questo percorso avviene nel modo seguente:

1. Il passo `child::ricetta[attribute::id='117']` seleziona il nodo `ricetta` con ID `'117'`.
2. Con il carattere `/`, *cambia il nodo contesto*: adesso, non è più `collezione`, bensì il nodo `ricetta` selezionato dal passo precedente.
3. Il passo `child::ingrediente` seleziona tutti gli ingredienti della ricetta 177.
4. Lo `/` indica che ciascuno di questi ingredienti diventa adesso il nodo contesto.
5. Per ogni nodo `ingrediente`, viene selezionato il valore dell'attributo `quant` (se presente). Questi valori costituiscono il risultato finale dell'intero percorso di locazione.

Come altro esempio, si considerano l'albero



(dove le lettere sono i nomi degli elementi, mentre i numeri servono solo a distinguere gli elementi con lo stesso nome), e il percorso

```
descendant::c/child::e/child::f
```

Partendo dal nodo contesto $\mathtt{a}_1$ (la radice):

1. `descendant::c` seleziona tutti i nodi `c` che sono discendenti (figli, o figli dei figli, ecc.) di $\mathtt{a}_1$: $\mathtt{c}_3$, $\mathtt{c}_8$, $\mathtt{c}_{12}$.
2. `child::e` seleziona i nodi `e` che sono figli dei nodi individuati dal passo precedente: $\mathtt{e}_4$, $\mathtt{e}_{13}$, $\mathtt{e}_{14}$

3. `child::f` seleziona i figli chiamati `f` dei nodi `e` trovati al passo precedente: $\texttt{f}_5$, $\texttt{f}_6$, $\texttt{f}_7$, $\texttt{f}_{15}$

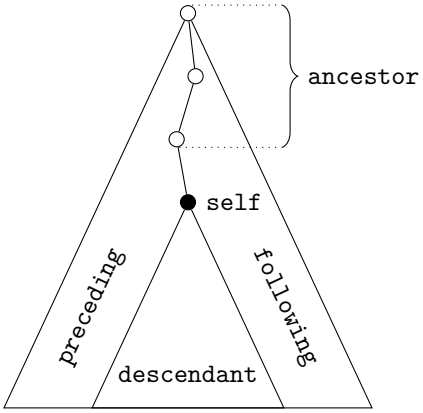
Complessivamente, il risultato del percorso è $\texttt{f}_5$, $\texttt{f}_6$, $\texttt{f}_7$, $\texttt{f}_{15}$. Si osserva che questo non è un albero (o documento) XML, ma solo una sequenza di nodi (estratti dall'albero di partenza). Questa è una differenza importante (già anticipata) rispetto al modello relazionale, nel quale il risultato di una query è *sempre* anch'esso una relazione.

2 Assi

In linea generale, un asse identifica una sequenza di nodi dell'albero che si trovano in una determinata posizione rispetto al nodo contesto. XPath (1.0) prevede 12 assi diversi:

- `child`: i figli del nodo contesto;
- `descendant`: i discendenti del nodo contesto;
- `parent`: il padre del nodo contesto (che è un singolo nodo, oppure nessuno, se il nodo contesto è la radice);
- `ancestor`: gli antenati del nodo contesto (il padre, il padre del padre, ecc., fino alla radice);
- `following`: tutti i nodi che, nel documento, appaiono dopo il nodo contesto, esclusi però i suoi discendenti;
- `preceding`: tutti i nodi che appaiono nel documento dopo il nodo contesto, esclusi i suoi antenati;
- `following-sibling`: i fratelli "a destra" del nodo contesto;
- `preceding-sibling`: i fratelli "a sinistra" del nodo contesto;
- `attribute`: gli attributi del nodo contesto (che *non* sono invece compresi negli altri assi elencati finora);
- `self`: il nodo contesto stesso;
- `descendant-or-self` e `ancestor-or-self`: assi "composti", che comprendono tutti i nodi `descendant` / `ancestor`, più il nodo contesto stesso.

Gli assi possono essere divisi in due categorie, secondo la *direzione* in cui si spostano nell'albero rispetto al nodo contesto: quelli che percorrono l'albero in avanti, e quelli che lo percorrono all'indietro. Inoltre, si osserva che, dato una qualsiasi nodo contesto, gli assi `self`, `ancestor`, `descendant`, `preceding` e `following` costituiscono una partizione di tutti i nodi dell'albero:



2.1 Abbreviazioni

Gli assi più utilizzati sono `child`, `descendant` e `attribute`. Perciò, esistono delle abbreviazioni che permettono di specificarli in modo più compatto:

- Se non si indica alcun asse, viene utilizzato l'asse `child` per default. Quindi, ad esempio, il percorso di locazione

  ```
  collezione/ricetta/ingrediente
  ```

  equivale a

  ```
  child::collezione/child::ricetta/child::ingrediente
  ```

- L'asse `attribute` può essere indicato con `@`: ad esempio, `@quant` è equivalente a `attribute::quant`.
- L'abbreviazione `//` indica l'asse `descendant-or-self`, cioè, in pratica, tutti i nodi del sottoalbero "corrente" (quello che ha come radice il nodo contesto).

3 Test di nodo

Un test di nodo agisce come un filtro per selezionare solo alcuni tipi di nodi tra quelli inclusi nell'asse specificato. I test di nodo più comuni sono:

- *nome*: seleziona tutti i nodi che hanno tale nome (se necessario, si può anche specificare un namespace: ad esempio, il passo di locazione `child::rcp:ricetta` indica i nodi figli che sono del tipo `ricetta`, definito nel namespace `rcp`);

- `*`: seleziona tutti i nodi della categoria corrispondente all'asse, cioè i nodi attributo per l'asse `attribute`, e i nodi elemento per tutti gli altri assi;
- `text()`: seleziona solo i nodi di testo.

4 Predicati

I predicati specificano ulteriori condizioni di selezione per filtrare i nodi individuati dall'asse e dal test di nodo.

Un aspetto importante è che un predicato può a sua volta contenere un altro percorso di locazione. Ad esempio, il percorso

```
descendant::ricetta[descendant::ingrediente[attribute::nome='zucchero']]
```

sempre riferito all'albero XML delle ricette, seleziona i nodi `ricetta` che comprendono lo zucchero tra gli ingredienti. Infatti, i passi di locazione scritti all'interno dei predicati *non cambiano* il nodo contesto, che, in questo caso, rimane sempre la radice, e perciò vengono appunto selezionati i nodi ricetta.

Scrivendo invece

```
descendant::ricetta/descendant::ingrediente[attribute::nome='zucchero']
```

al secondo passo il nodo contesto diventerebbe la `ricetta`, e sarebbero allora stati selezionati i nodi `ingrediente` aventi l'attributo `nome="zucchero"`. Quindi, questi due percorsi hanno risultati *fondamentalmente diversi*.